Discrete Mathematics 111 (1993) 361–366
North-Holland

The cyclic compact open-shop scheduling problem

N.V.R. Mahadev
Northeastern University, Boston, MA, USA

Ph. Solot and D. de Werra
Ecole Polytechnique Fédérale de Lausanne, Switzerland

Received 22 July 1991

1. Introduction

We consider a variation of the classical nonpreemptive open-shop scheduling problem. For the normal open shop, finding whether a schedule exists in T time units is NP-complete [2].

The main difference with the classical model is that we will consider so-called periodical or cyclic schedules; this means that the schedule obtained is repeated periodically. If the length of a period is k time units, then a job which is on a processor during the first p time units of a period and the last q time units of a period will be considered as being processed without preemptions on that processor during $q+p$ consecutive time units.

Such scheduling problems are sometimes called *cylindrical* (instead of *cyclic*); this name stems from the fact that we take the processor schedules (and the job schedules) for one period and make a cylinder by gluing together the $t=0$ and $t=k$ axes.

Cyclic problems occur more and more frequently in automated production systems; some types of cyclic problems are discussed in [3].

Another difference from the classical models is that we will consider compact schedules only.

This additional requirement is sometimes encountered in production systems. A job when started must never wait during its processing (some operations need, for instance, to be performed at a fixed temperature) . In a similar way, when a processor starts working, there should be no interruption (because starting the processor may be a long or difficult task or it may consume much energy).

Correspondence to: D. de Werra, Ecole Polytechnique Fédérale de Lausanne, Département de Mathématiques, MA (Ecublens), CH-1015 Lausanne, Switzerland.

0012-365X/93/$06.00

In a cyclic schedule, this means that if we consider the cylinder representing the job (processor) schedules in each period, each job (processor) will have at most one idle time interval on the cylinder.

No-wait schedules (i.e. compact schedules) have been considered in [5]; they are also NP-complete. We will give a complete formulation of the cyclic compact open-shop scheduling problem in the next section and derive a graph-theoretic model. A characterization in terms of graphs will be given for problems which have cyclic schedules with some additional requirements. The complete derivation of these results can be found in [6].

All graph-theoretic terms not defined here can be found in [1].

2. Formulation of the CCOSS

The cyclic compact open-shop scheduling problem (CCOSS) is defined as follows: a collection $\mathscr{P}$ of processor types $P_1, \ldots, P_m$ is given together with a set $\mathscr{J} = \{J_1, \ldots, J_n\}$ of jobs. Each job J_j must be processed during p_{ij} consecutive time units on a processor of type P_i. We shall assume that all p_{ij}'s are integral.

Each job can be on at most one processor at a time and, similarly, each processor can work on at most one job at a time. There is no order in which a job must visit the processors.

We are given a period of k time units within which jobs $J_1, \ldots, J_n$ must be completed. The schedule is then repeated in a periodic way.

Clearly, if k is too small, one has to process in parallel several copies of jobs J_j for which $\sum_i p_{ij} > k$; similarly, several processors of type P_i have to be used if $\sum_j p_{ij} > k$.

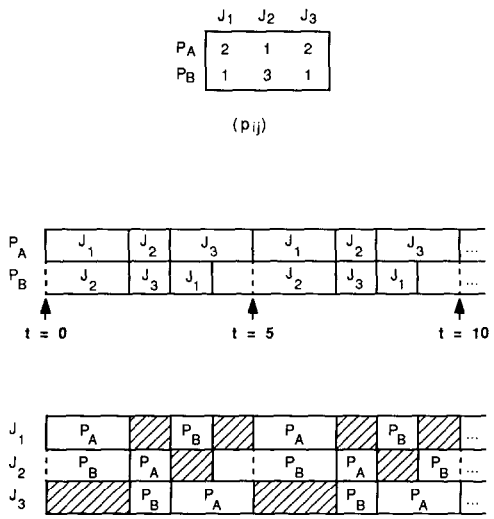
We require furthermore that the processing schedule of each job is compact, i.e. once a job starts being processed, it visits continuously the processors to which it is assigned.

Similarly, processors work continuously when they start working in a period.

The smallest possible number of processors of type P_i is $\lceil \sum_j p_{ij}/k \rceil$ and the smallest possible number of copies of job J_j which must be simultaneously in process during a period in order to meet the production requirement is $\lceil \sum_i p_{ij}/k \rceil$. Generally, we cannot, for a given k, find a production schedule which is compact as requested and which uses the absolute minimum number of processors of each type and the absolute minimum number of simultaneous copies of each job J_j.

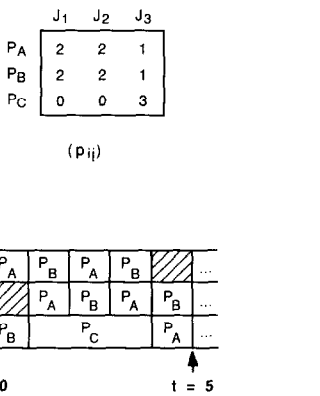
Fig. 1 represents a matrix of processing times p_{ij} for a problem with two processor types P_A, P_B and three jobs J_1, J_2, J_3. We take $k = 5$; so, only one copy of each job is needed in each period and only one processor of each type. A periodic schedule is given in Fig. 1; note that it is not feasible since for job J_1 there are interruptions (idle times) in the time intervals [2, 3] and [4, 5].

Fig. 2 shows another example with $k = 5$ where 3 processors (one of each type) and 3 jobs are present. This schedule is compact, but it is not feasible since for jobs J_1



A periodic schedule (not compact for J_1) with period k = 5.

Fig. 1.



A periodic schedule (compact for all jobs and for all processors) with period k = 5 and with preemptions for jobs J_1 and J_2.

Fig. 2.

and J_2 there are preemptions. In fact, for this example there is no cyclic compact schedules (without preemptions).

Our purpose is to characterize the cases where such compact schedules exist for any choice of the length k of a period.

3. A graph-theoretic model

An instance of the CCOSS is given by an $m \times n$ matrix $P=(p_{ij})$ and a positive integer k (length of the production period).

We may associate with each instance a bipartite graph $G=(\mathscr{P}, \mathscr{J}, E)$; for each entry $p_{ij}>0$ of P we introduce an edge $e=[P_i, J_j]$ with weight $c_e=p_{ij}$.

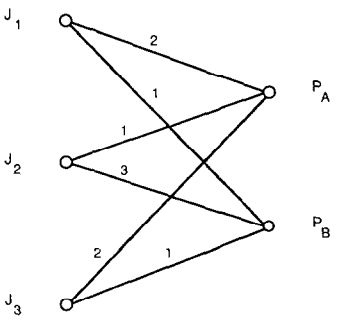
The existence of a schedule of the above type corresponds to a special type of coloring of the edges of G. Let $K=\{1, \ldots, k\}$ be a cyclically ordered set; this means that color 1 follows color k. A *cyclic pseudointerval* (cpi) of length q in K is a sequence of q consecutive integers in K; note that if $q>k$ then the same integer may occur several times in a cpi.

Given a bipartite graph $G=(\mathscr{P}, \mathscr{J}, E, c)$, where each edge e has an integral weight $c_e \geqslant 0$, a *cpi k-coloring* of G is an assignment of a cpi S_e of length c_e to each edge e such that in each bundle $B(x)$ of edges adjacent to the same node x the cpi's S_e can be combined into a cpi. (They can be linearly ordered in such a way that, if the last color of an S_e is a, the first color of the next $S_{e'}$ is $a+1$.)

An example of cpi k-coloring for a bipartite graph G and an associated schedule are given in Figs. 3–5. Fig. 3 shows a graph $G=(\mathscr{P}, \mathscr{J}, E)$ with weights $c_e = p_{ij}$ on each edge $e=[P_i, J_j]$. A cpi 3-coloring is shown in Fig. 4. An associated schedule is given in Fig. 5: note that one needs two processors of type P_A and two of type P_B. Two copies of J_2 have to be processed during each period.

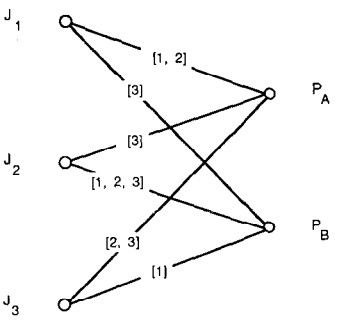
A graph $G=(\mathscr{P}, \mathscr{J}, E, c)$ is *cpi-perfect* if for any k and for any choice of nonnegative integral weights c_e for the edges e in E there exists a cpi k-coloring of G.

In terms of scheduling, cpi-perfect graphs correspond to sets C of entries in P (we call these sets *configurations*) such that for any assignment of nonnegative integer weights to the entries of C there exists a schedule in k time units. A *mouth* is



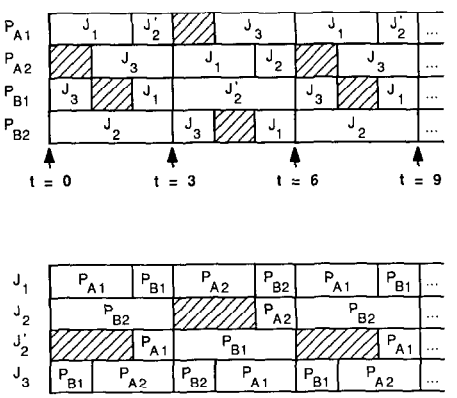
The graph G with weight w_e on each edge e

Fig. 3.



A cpi 3-coloring of G

Fig. 4.



A periodic schedule with period k = 3 (for the same data as in Fig. 1)

Fig. 5.

a collection of three-node disjoint chains (of same parity) linking two nodes. The main characterization of cpi-perfect graphs is given by the following proposition.

Proposition 3.1 (de Werra et al. [6]). *For a connected bipartite graph G the following statements are equivalent:*

(a) *G is cpi-perfect.*

(b) *G is outerplanar.*

(c) *G does not contain a mouth as a partial subgraph.*

Remark. Replacing bundles by (inclusionwise) maximal sets of mutually adjacent edges, one could define cpi-perfection for general graphs. It turns out that the only non-bipartite connected graphs which are cpi-perfect are triangles. We can therefore restrict our attention to the bipartite graphs.

Note that for $k = \max\{\max_i \sum_j p_{ij}, \max_j \sum_i p_{ij}\}$ there exists a schedule with a period of k time units; one processor of each type P_i needed and one copy of each job J_j is in process during each period.

There is a simple coloring algorithm for constructing a cpi k-coloring in a cpi-perfect graph G; it is based on a property of a plane embedding of G.

Every 2-connected component of G can be obtained by starting from a cycle C and by repeatedly applying the following rule:

> choose an edge e (which has not been chosen before) and introduce a handle (an odd chain with the same endpoints as e and with new intermediate nodes). It is then easy to color the edges in the same order as they are introduced. The detailed algorithm is described in [6]; 2-connected components are colored one after the other by choosing one which has a node in common with a colored 2-connected component.

4. Additional remarks

An edge k-coloring of a graph G is called *equitable* if at any node x the number of occurrences of the different colors on edges adjacent to x differ from one another by at most one.

So, cpi k-coloring are equitable k-colorings, but an arbitrary equitable k-coloring may not be a cpi k-coloring.

In our CCOSS problem we could consider only the case where $k = k(P) = \max\{\max_i \sum_j p_{ij}, \max_j \sum_i p_{ij}\}$ and ask: What are the configurations (graphs) for which a schedule exists with a period of $k(P)$ time units (for this $k = k(P)$ only), whatever the choice of values p_{ij}? In such a case, no duplication of jobs and processors is needed.

It turns out that this gives exactly the class of cpi-perfect graphs (see [6]).

Furthermore, relaxing the compactness requirements for jobs and/or for processors should give a wider class of graphs than cpi-perfect graphs.

Our attempts to characterize such graphs have given only partial results.

Suppose we restrict the weights c_e of all edges e to $\{0, 1\}$ and we relax the compactness requirements. If we require only that an edge k-coloring exists for $k =$ maximum weight of a set of mutually adjacent edges, then we get line-perfect graphs, i.e. graphs whose line-graphs are perfect. A characterization of these follows from the results on claw-free graphs in [4]. Cpi-perfect graphs are a subclass of line-perfect graphs.

References

[1] C. Berge, Graphes (Gauthiers-Villars, Paris, 1983).
[2] J. Blazewicz, W. Cellary, R. Slowinski and J. Weglarz, Scheduling under Resource Constraints: Deterministic Models (Baltzer AG, Basel, 1986).
[3] W. Dauscha, H.D. Modrow and A. Neumann, On cyclic sequence type for constructing cyclic schedules, Z. Oper. Res. 29 (1985) 1–30.
[4] K.R. Parthasaraty and G. Ravindra, The strong perfect graph conjecture is true for $K_{1,3}$-free graphs, J. Combin. Theory Ser. B 21 (1976) 212–223.
[5] S.S. Reddi and C.V. Ramamoorty, On the flowshop sequencing problem with no wait in process, Oper. Res. Quart. 23 (1972) 323–331.
[6] D. de Werra, N.V.R. Mahadev and Ph. Solot, Periodic compact scheduling, to appear.